

第四届 (2022) 集成电路 EDA 设计精英挑战赛

赛题指南

- 一、 **赛题名称：**数字集成电路动态功耗优化策略分析与评估
- 二、 **命题企业：**芯华章科技股份有限公司
- 三、 **赛题 Chair：**黄乐天 电子科技大学、林亦波 北京大学
- 四、 **赛题描述**

作为衡量数字芯片设计的重要的指标 (PPA) 之一, 低功耗设计的分析与优化显得尤为重要。对于 CMOS 电路而言, 功耗主要由动态功耗和静态功耗两部分组成。静态功耗通常由漏电流引起, 而动态功耗则主要由开关功耗, 短路功耗两部分组成。图 1 给出了典型的 CMOS 反相器的动态功耗构成原因说明。

开关功耗即电路在开关过程中对输出节点的负载电容 C_L 充放电所引起的功耗。具体的功耗数值可通过公式 $P_{switch} = \frac{1}{2} V_{DD}^2 \times C_L \times T_r$ 进行计算。其中 V_{DD} 为供电电压, T_r 为输入信号翻转率 (单位时间内信号的翻转次数), C_L 为后级电路等效的电容负载大小。由此可见, 开关功耗与电路的供电电压值的平方成正比, 减小供电电压可以有效降低开关功耗。因此在低功耗设计中, 在保证电路功能的前提下, 可以对不同的电路模块提供不同的供电电源或者供电电压 (MSMV)。开关功耗与信号翻转率成正比, 减少不必要的信号翻转是降低功耗的有效手段之一, 比如通过优化 RTL 编码风格以减少状态转换过程中的信号翻转次数, 如 Gray-Code 编码等。更为常见的如门控时钟技术 (Clock-Gate), 可以给特定的触发器添加时钟门控, 可以有效减少触发器

上时钟信号的翻转，而时钟信号的翻转恰恰是最频繁的。总而言之，可以通过优化信号翻转次数进而达到功耗优化目的。

短路功耗是出现在输入信号翻转时，因为信号的翻转不会在瞬间完成进而导致 PMOS 和 NMOS 总是存在同时导通情形，PMOS 和 NMOS 的同时导通形成电源到地线之间的短路电流。公式 $P_{sc} = I_{sc} \times V_{DD} = \frac{1}{12} \times k \times \tau \times f_{clk} \times (V_{DD} - 2V_{th})^3$ 给出了计算短路功耗的一种方法。其中 k 为介电常数， τ 为波形上升沿与下降沿时间， f_{clk} 为时钟频率， V_{DD} 为供电电压， V_{th} 为器件的阈值电压。

相比开关功耗，通常短路功耗在动态功耗中比例较小。因此分析与评估动态功耗优化策略主要针对影响开关功耗的因素进行，更进一步地，本赛题主要针对不同功耗优化策略下的信号翻转信息进行分析，进而对电路的动态功耗策略做定性的评估。

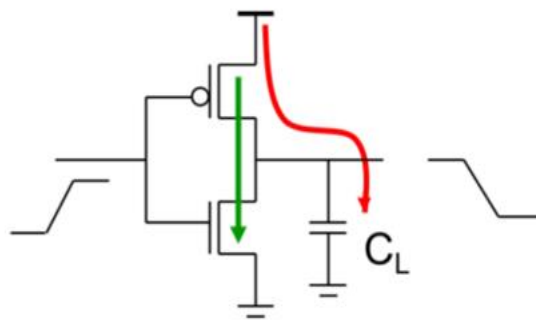


图 1 CMOS 反相器的动态功耗构成示意图

VCD (Value Change Dump) 文件是 IEEE1364 标准 (Verilog 语言标准) 中定义的一种通用格式的 ASCII 文件。它主要包含了头信息，变量的定义以及变量值的变化信息等。VCD 完整的记录整个仿真过程中信号的变化信息，因此 EDA 工具可以通过 VCD 文件来评估设计中的功耗。

对于下述 RTL Verilog 和 Testbench (测试平台), 通过仿真工具生成类似如下的 VCD 文件。

| | |
|--|---|
| <pre>RTL: module top(clk, reset); input clk, reset; wire[3:0] out; sub sub_i(.clk(clk), .reset(reset), .out(out)); endmodule module sub(clk, reset, out); input clk, reset; output out; reg[3:0] out; always @(posedge clk) begin out <= (reset)? 0 : out + 1; end endmodule</pre> | <pre>Testbench: `timescale 1ns/1ns module test(); reg clk, reset; always #50 clk = ~clk; initial begin clk = 0; reset = 1; #100 reset = 1; #400; \$finish; end top top(clk, reset); endmodule</pre> |
|--|---|

```
$scope module test $end
$scope module top_i $end
$var wire 1 n1 clk $end
$var wire 1 n2 reset $end
$var wire 4 n3 out [3:0] $end
$scope module sub_i $end
$var wire 1n2 reset $end
$var wire 1 n1 clk $end
$var reg 4 n3 out [3:0] $end
$upscope $end
$upscope $end
$upscope $end
$enddefinitions $end
#0
bxxxx n3
0 n1
1 n2
#50
1 n1
b0000 n3
#100
0 n1
0 n2
#150
1 n1
b0001 n3
#200
0 n1
#250
1 n1
b0010 n3
#300
0 n1
#350
1 n1
b0011 n3
#400
0 n1
#450
1 n1
b0100 n3
#500
0 n1
```

为更加形象具体的展示 VCD 文件的细节，图 2 给出了芯华章科技股份有限公司的商业调试软件 Fusion Debug 的波形展示图。这里可以明显看出在整个仿真阶段 (0ns-500ns)，模块 top 中的信号有如下的翻转行为（包括翻转次数 tc，信号逻辑值为 1 的时间 t1，信号逻辑值为 0 的时间 t0，信号逻辑值为 x 的时间 tx，以及信号逻辑值为 1 的概率又称静态概率 sp）：

top.clk: tc = 10; t1 = 250ns; t0 = 250ns; tx = 0ns; sp = 0.5

top.reset: tc = 1; t1 = 100ns; t0 = 400ns; tx = 0ns; sp = 0.2

top.out[3]: tc = 0; t1 = 0ns; t0 = 450ns; tx = 50ns; sp = 0.0

top.out[2]: tc = 1; t1 = 50ns; t0 = 400ns; tx = 50ns; sp = 0.1

top.out[1]: tc = 2; t1 = 200ns; t0 = 250ns; tx = 50ns; sp = 0.4

top.out[0]: tc = 4; t1 = 200ns; t0 = 250ns; tx = 50ns; sp = 0.4

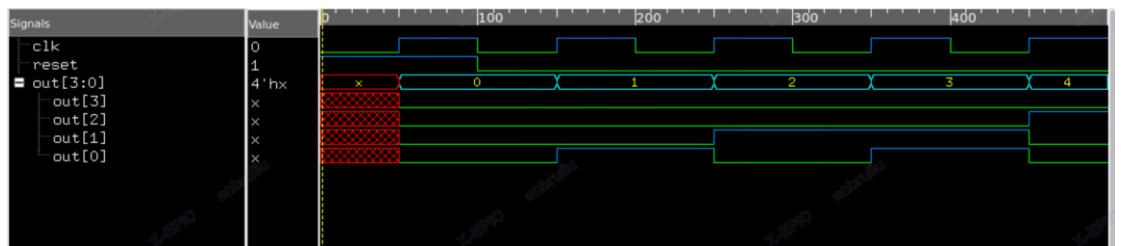


图 2 信号变化波形图 (芯华章科技有限公司-Fusion Debug 提供)

具体要求如下：

- a. 实现 VCD 文件的解析工具，并能建立简单的模型；
- b. 基于 VCD 文件可以提供设计中信号的翻转情况统计；

五、 评分标准

评分标准包括功能覆盖、代码质量以及代码性能三部分内容

1.1 功能覆盖 (40 分)

- 1) 支持 VCD 文件解析与建模 (5 分)

2) 支持信号翻转次数以及翻转率提取 (35 分)

a.支持在整个仿真时间内的全部信号翻转次数以及翻转率提取 (15 分)

具体要求:

针对上述事例, 应输出下述内容到文件 summary.csv

| | | | | | |
|------------------|---------|---------------|---------------|--------------|-------------|
| top.clk | tc = 10 | t1 = 250ns | t0 = 250ns | tx = 0ns | sp = 0.5 |
| top.reset | tc = 2 | t1 = 400ns | t0 = 100ns | tx = 0ns | sp = 0.8 |
| top.out[3] | tc = 0 | t1 = 0ns | t0 = 450ns | tx = 50ns | sp = 0.0 |
| top.out[2] | tc = 1 | t1 = 50ns | t0 = 400ns | tx = 50ns | sp = 0.1 |
| top.out[1] | tc = 2 | t1 = 200ns | t0 = 250ns | tx = 50ns | sp = 0.4 |
| top.out[0] | tc = 4 | t1 = 200ns | t0 = 250ns | tx = 50ns | sp = 0.4 |
| top/sub_i.clk | tc = 10 | t1 = 250ns | t0 = 250ns | tx = 0ns | sp = 0.5 |
| top/sub_i.reset | tc = 2 | t1 = 400ns | t0 = 100ns | tx = 0ns | sp = 0.8 |
| top/sub_i.out[3] | tc = 0 | t1 = 0ns | t0 = 450ns | tx = 50ns | sp = 0.0 |

| | | | | | |
|------------------|--------|-------|-------|------|------|
| | | t1 = | t0 = | tx = | sp = |
| top/sub_i.out[2] | tc = 1 | 50ns | 400ns | 50ns | 0.1 |
| | | t1 = | t0 = | tx = | sp = |
| top/sub_i.out[1] | tc = 2 | 200ns | 250ns | 50ns | 0.4 |
| | | t1 = | t0 = | tx = | sp = |
| top/sub_i.out[0] | tc = 4 | 200ns | 250ns | 50ns | 0.4 |

b.支持指定模块下信号的翻转次数以及翻转率提取（10分）

具体要求：

在指定 scope 的条件下，仅给出对应 scope 下的信号翻转信息。如上述事例，运行程序“-scope top/sub_i”时，应输出下述内容到文件

summary.csv：

| | | | | | |
|------------------|---------|----------|-------|------|------|
| | | t1 = | t0 = | tx = | sp = |
| top/sub_i.clk | tc = 10 | 250ns | 250ns | 0ns | 0.5 |
| | | t1 = | t0 = | tx = | sp = |
| top/sub_i.reset | tc = 2 | 400ns | 100ns | 0ns | 0.8 |
| | | | t0 = | tx = | sp = |
| top/sub_i.out[3] | tc = 0 | t1 = 0ns | 450ns | 50ns | 0.0 |
| | | t1 = | t0 = | tx = | sp = |
| top/sub_i.out[2] | tc = 1 | 50ns | 400ns | 50ns | 0.1 |
| | | t1 = | t0 = | tx = | sp = |
| top/sub_i.out[1] | tc = 2 | 200ns | 250ns | 50ns | 0.4 |

| | | | | | |
|------------------|--------|-------|-------|------|------|
| | | t1 = | t0 = | tx = | sp = |
| top/sub_i.out[0] | tc = 4 | 200ns | 250ns | 50ns | 0.4 |

c.支持指定时间窗口内全部信号的翻转次数以及翻转率提取（10分）

具体要求：

在指定时间窗口条件下，仅给出对应时间范围内的信号翻转信息。如上述事例，运行程序“-begin 250ns -end 400ns”，应输出下述内容到文件

summary.csv：

| | | | | | |
|---------------|--------|------------|------------|----------|---------------|
| | | t1 = | t0 = | tx = | sp = |
| top.clk | tc = 3 | 100ns | 50ns | 0ns | 0.666667 |
| top.reset | tc = 0 | t1 = 0ns | t0 = 150ns | tx = 0ns | sp = 0.0 |
| top.out[3] | tc = 0 | t1 = 0ns | t0 = 150ns | tx = 0ns | sp = 0.0 |
| top.out[2] | tc = 0 | t1 = 0ns | t0 = 150ns | tx = 0ns | sp = 0.0 |
| top.out[1] | tc = 0 | t1 = 150ns | t0 = 0ns | tx = 0ns | sp = 1.0 |
| top.out[0] | tc = 1 | t1 = 50ns | t0 = 100ns | tx = 0ns | sp = 0.333333 |
| top/sub_i.clk | tc = 3 | t1 = 100ns | t0 = 50ns | tx = 0ns | sp = 0.666667 |

| | | | | | |
|------------------|--------|---------------|---------------|-------------|------------------|
| top/sub_i.reset | tc = 0 | t1 = 0ns | t0 = 150ns | tx = 0ns | sp = 0.0 |
| top/sub_i.out[3] | tc = 0 | t1 = 0ns | t0 = 150ns | tx = 0ns | sp = 0.0 |
| top/sub_i.out[2] | tc = 0 | t1 = 0ns | t0 = 150ns | tx = 0ns | sp = 0.0 |
| top/sub_i.out[1] | tc = 0 | t1 = 150ns | t0 = 0ns | tx = 0ns | sp = 1.0 |
| top/sub_i.out[0] | tc = 1 | t1 = 50ns | t0 = 100ns | tx = 0ns | sp = 0.333333 |

1.2 代码质量 (10分)

- 1) 设计测试回归系统用于对整个测试集进行回归测试 (10分)

1.3 代码性能 (50分)

对于功能正确且完整的作品，以 compile time 以及 peak memory 为性能衡量依据。

- 1) 提供用以测试性能的 profile data (提供 time 以及 peak memory 信息) (10分)

- 2) 单核 (单进程) 模式下性能评分标准 (25分)

以 time 为主要衡量依据，性能超越 80% 参赛队伍得 15 分，性能超越 50% 参赛队伍得 10 分，其他队伍不得分

以 peak memory 为衡量依据，在 time 相差 5% 以内的参赛队伍中内存消耗较少的队伍可得 10 分

3) 多核（四核）模式下性能评分标准（15 分）

以 time 为主要衡量依据，性能超越 80% 参赛队伍得 10 分，性能超越 50% 参赛队伍得 5 分，其他队伍不得分

以 peak memory 为衡量依据，在 time 相差 5% 以内的参赛队伍中内存消耗较少的队伍可得 5 分

1.4 附加分（10 分）

提供程序可以绘制出随时间变化信号翻转统计图，图 3 给出了基于芯华章科技股份有限公司商业调试软件 Fusion Debug 提供的信号翻转信息图。

横轴坐标为时间，纵轴坐标为该时刻翻转的信号数量。

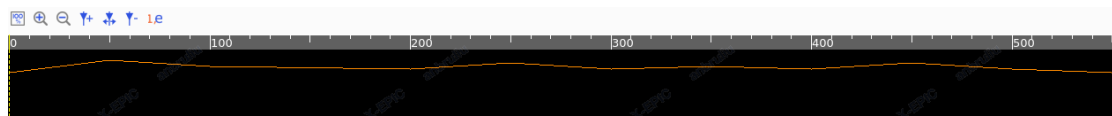


图 3 信号翻转统计图（芯华章科技有限公司-Fusion Debug 提供）

六、 测试集

比赛开始时，选手自行设计测试用例来检测自己所设计功能模块的正确性。验收时，将提供一个标准测试集（10-20 个 cases）用以测试选手所提交的软件，作为功能覆盖率的打分标准。另外，会有一个性能测试集（10 个左右 benchmark），作为性能的打分标准。

七、 技术指导

对于不熟悉 Verilog, VCD 文件格式以及低功耗概念的参赛者，可以提供一定的技术指导，使参赛者可以尽快进入核心功能开发阶段。

八、 参考资料

- 1) 1801-IEEE Standard for Design and Verification of Low-Power, Energy-Aware Electronic Systems
- 2) 1364-2001 - IEEE Standard Verilog Hardware Description Language